Integrated volume rendering and data analysis in wavelet space

Integrated Volume Rendering and Data Analysis in Wavelet Space

M. H. Gross
Computer Science Department
Swiss Federal Institute of Technology
(ETH–Zürich)
Internal Report No. 250
September 24, 1996

Abstract

The following paper describes a framework for volume data analysis and visualization using wavelet transforms (WTs). It bases on the idea that on the one hand WTs have approved to provide powerful features for various applications in the field of data analysis. Due to the basic properties of this transform, such as local support and orientation selectivity, many researchers tried to exploit the WT for the extraction of local data features. In particular, in image texture analysis, wavelet based feature extractors accomplished highly accurate segmentation results that can be extended straightforwardly to volumetric data sets.

On the other hand, due to the compact coding properties of orthonormal wavelets the WT allows to decompose any finite energy function and to approximate it from its bases. Therefore we can develop rendering methods, that provide an approximate solution of the low albedo volume rendering equation. Since generally, the wavelets are not given in a closed form, the approach reported in this paper bases on a piecewise polynomial representation. Isosurfaces can easily be obtained from the data either by ray tracing of the bases or by simple marching cubes techniques.

Hence, the WT provides a uniform data representation that features both data analysis and data visualization. The paper introduces first to the mathematical foundations of the WT, it reviews briefly different types of basis functions and it stresses implementation details using iterated QMF–pair filters. Furthermore, separable extensions to multiple dimensions are explained and it is elucidated, how local data features can be derived from the wavelet pyramid. For data analysis purposes, a newly developed image texture analysis pipeline contains a WT, a principal component analysis, normalization procedures and a neural network. Volume rendering is accomplished by projecting the 3D wavelets onto the viewing ray and by piecewise analytic integration of the rendering equation. The methods reported here are illustrated by various examples.

1 Introduction

The wavelet–transform has gained much attraction for getting innovative solutions to various technical problems in a broad range of applications. Following the mathematical formulation of the wavelet–transform ([12], [5], [24]) a lot of work has been done to employ wavelets for hierarchical data coding and representation. This paper, however reports on a framework for integrated volume rendering and data analysis in wavelet space. Here, the properties of the WT are exploited to derive local data features that allow to solve segmentation and classification tasks. Consequently, the WT provides a unique data representation that enables to incorporate data analysis as an important preprocessing step in rendering. This new concept reported below bases on the following observations:

In image processing, research has been focussed on the investigation of multiresolution analysis techniques for texture feature extraction. [3] used Gabor–functions of different spectral ranges and orientations to derive a multiscale sight onto the texture. [33] employed a set of local linear transforms as an initial step in a texture recognition pipeline and combined them with dimensionality reduction techniques. Orthogonal wavelets and wavelet packages for texture analysis had been introduced by [19] and [4]. Obviously, most approaches derive global wavelet features in terms of means from a specific type of texture and they transform each subset separately into the wavelet space. This procedure renders high classification rates but it neither takes into account texture boundaries nor does it take advantage of the spatial localization of the wavelet transform. Consequently, for many practical applications, as for instance remote sensing or medical imaging, difficulties will arise in applying these techniques because they don't provide large spatial coherent textures. [9] and [11] describe a different approaches to derive features from wavelet–transforms using the spatial and frequency localization of the WT. Here, the segmentation pipeline consists of WT, principal component analysis, normalization and a neural classifier. This method can easily be extended to volume textures.

In volume rendering, initial work has also been done to employ wavelets for data representation. [26] describes a way to get isosurfaces from volume data by computing a continuous approximation with wavelet bases. [13] used wavelets to control elegantly volume morphing algorithms and [22] considers line integration as a texture splatting problem. [8] describes a new method to approximate the volume rendering equation using wavelet transforms. For this purpose, the initial volume data set is transformed into wavelet space using separable 3D extensions of orthonormal wavelet types. Since some wavelets, such as the Daubechies, are not given in a closed form solution they approximate the basis functions with piecewise polynomial splines. This allows a continuous hierarchical approximation of the 3D data set. Due to the local support of the wavelet bases, the local level–of–detail can be controlled efficiently. This enables to emphasize local features of interest. Once this continuous piecewise polynomial approximation is computed, the volume intensity function along the ray can easily be formulated and a linear approximation of the exponential absorption term provides a polynomial approximation of the entire rendering integral. This finally leads towards analytic solutions.

In order to define an integrated framework, this paper reviews and unifies the initial concepts provided by [10] and [9]. The organization of the paper is as follows: First of all, the mathematical basics of the wavelet–transform are briefly elucidated and it is shown, how to get sepa-

rable 2D and 3D extensions. Different wavelet types are discussed and compared to each other in terms of smoothness and compact support. The second chapter introduces to the concepts of integrating volume rendering and data analysis in wavelet space. The data analysis pipeline is illuminated in detail in chapter 3 and examples are given for the segmentation performance of image textures. Chapter 4 sheds light on the continuous approximation of the data with wavelet bases and provides an approximation of the volume rendering integral for the ray–casting process. It is illustrated how to compute isosurfaces in wavelet space and compares results obtained from Kalra's [17] method are compared to those of the marching cubes [23] reconstruction techniques using examples from laser range data sets.

2 Mathematical Foundations

2.1 The Wavelet Transform

The wavelet–transform (WT) is an integral transform of any finite energy function $f(x) \in L^2(\mathbf{R})$ using a set of similar basis functions $\psi_{ab}(x)$. Its generic continuous form description for real functions is provided as the following inner product:

$$WT_{f,\psi}(a,b) = \langle f, \psi_{ab} \rangle = \int_{-\infty}^{\infty} \psi_{ab}(x) f(x) dx \qquad a,b \in \mathbf{R} \tag{1}$$

$L^2(\mathbf{R})$ denotes the Hilbert space of square integrable functions.

The individual basis functions on the real axis are derived from each other by scaling and shifting one prototype function ψ controlled by the parameters a and b respectively [12].

$$\psi_{ab}(x) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{x-b}{a}\right) \tag{2}$$

One required property of the orthonormal bases is their band–pass behavior which is defined as

$$\Psi_{ab}(0) = 0, \ \Psi_{ab}(\omega) : \textit{Fourier Transform of } \psi_{ab}(x) \tag{3}$$

Like any other type of linear transform the WT enables the decomposition and the expansion of the initial function $f(x)$ by linear combinations of the basis functions.

In order to handle this method with a computer, it is necessary to set up a discrete version. A dyadic scaling of the bases with $a = 2^m$ and a unit shift $b = p\,2^m$ yields:

$$\psi_{mp}(x) := 2^{-\frac{m}{2}} \psi(2^{-m}x - p) \qquad , \ \left(\psi_{mp}\right)_{p \in \mathbf{Z}} \textit{ basis of vectorspace } U_m \tag{4}$$

m: 1,..,M denotes the depth of the iteration.

In many construction schemes, the bases are furthermore supposed to be orthonormal to each other [5].

$$\langle \psi_{mp}, \psi_{\tilde{m}\tilde{p}} \rangle = \int_{-\infty}^{\infty} \psi_{mp}(t)\psi_{\tilde{m}\tilde{p}}(t)dt = \delta_{m\tilde{m}}\delta_{p\tilde{p}} \tag{5}$$

$$\text{with} \quad \delta_{ij} := \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases} \qquad (Kronecker - Delta - function)$$

Thus, the transform with discrete orthonormal wavelets can be formalized as

$$DWT_{f,\psi}(m,p) = 2^{-\frac{m}{2}} \int_{-\infty}^{\infty} \psi(2^{-m}x - p) \cdot f(x)\, dx \qquad m,p \in \mathbf{Z} \tag{6}$$

Mallat [24] stresses this concept by defining a set of multiresolution function spaces V_m that provide an approximations of all $f(x) \in L^2(\mathbf{R})$. The space of resolution m is derived from the higher resolution step by adding the orthogonal complement space U_m.

$$V_{m-1} = V_m \oplus U_m \tag{7}$$

$\oplus$: *direct sum*

It can be proofed that the so–called scaling functions $\phi \in L^2(\mathbf{R})$ with $\phi_{mp}(x) = 2^{-\frac{m}{2}}\phi(2^{-m}x - p)$ provide orthonormal bases of the vectorspaces V_m in each resolution step. The wavelet $\psi \in L^2(\mathbf{R})$ with $\psi_{mp}(x) = 2^{-\frac{m}{2}}\psi(2^{-m}x - p)$ however is proofed to be a basis of the orthogonal complement space U_m.

The statements explained above offer an iterated decomposition scheme, where an initial discrete function can successively be approximated for a given iteration depth M using the scaling function and wavelets in each orthogonal complement space U_m.

$$V_0 = V_M \oplus U_M \oplus \cdots \oplus U_1 \tag{8}$$

Hence, we obtain the following approximation:

$$\begin{aligned} f(x) := \sum_p c_p^0 \phi_{0p}(x) &= \sum_p c_p^1 \phi_{1p}(x) + \sum_p d_p^1 \psi_{1p}(x) \qquad f(x) \in V_0 \\ &= \sum_p c_p^M \phi_{Mp}(x) + \sum_{m=1}^{M} \sum_p d_p^m \psi_{mp}(x) \end{aligned} \tag{9}$$

c_p^M and d_p^m denote the coefficients of the transform.

2.2 Extensions to Multiple Dimensions

So far, the definitions were restricted to the one–dimensional case. For multidimensional signal processing, as image analysis or volume rendering, it is necessary to extend the method to multiple dimensions. Beside of the non–trivial nonseparable case [31] there is a straightforward way to accomplish this by means of non–standard tensor product extensions of the one–

dimensional representatives of the function spaces and of their bases. Once V_m is given, we can define a 2D version V_m^2 by

$$\begin{aligned} V_m^2 &= V_m \otimes V_m \\ V_m^2 &= \left(V_{m+1} \oplus U_{m+1}\right) \otimes \left(V_{m+1} \oplus U_{m+1}\right) \\ V_m^2 &= V_{m+1}^2 \oplus U_{m+1}^{2,1} \oplus U_{m+1}^{2,2} \oplus U_{m+1}^{2,3} \end{aligned} \qquad (10)$$

$\otimes$: *Tensor product operator*

In eq. 10 the initial space is broken up into 3 spaces that account for the differences in the signal and into a low–pass part. The resulting 2D versions of the respective wavelet bases yield:

$$\begin{aligned} \phi_{mpq}^2(x,y) &:= 2^{-m}\phi(2^{-m}x - p)\phi(2^{-m}y - q) \\ \psi_{mpq}^{2,1}(x,y) &:= 2^{-m}\psi(2^{-m}x - p)\phi(2^{-m}y - q) \\ \psi_{mpq}^{2.2}(x,y) &:= 2^{-m}\phi(2^{-m}x - p)\psi(2^{-m}y - q) \\ \psi_{mpq}^{2.3}(x,y) &:= 2^{-m}\psi(2^{-m}x - p)\psi(2^{-m}y - q) \end{aligned} \qquad (11)$$

Unfortunately, some wavelet types are not defined in a closed form solution and consequently the convolution products of eq. (1) cannot be computed explicitly. The implementation of orthogonal wavelet transforms often employs so–called quadrature mirror pair filters (QMFs). The basic scheme of this filter bank is illustrated in fig. 1a for a decomposition of 2D data sets. The initial data set is filtered along the x– and y– axis and subsampled by the factor 2 using the two filters $H'(\omega) = \sum_{n=-\infty}^{\infty} h(-n)e^{in\omega}$ and $G'(\omega) = e^{-i\omega}\overline{H'(\omega + \pi)}$ with $h(n) := 2^{-\frac{1}{2}} \int_{-\infty}^{\infty} \phi\left(\frac{x}{2}\right)\phi(x - n)dx$ respectively. The result of this process are three detail signals $D_{2^m}{}^1 f, \cdots, D_{2^m}{}^3 f$ that account for the oriented wavelets in this channel and a low pass signal $A_{2^m} f$ that is decomposed further. This iterated scheme corresponds to a dyadic subband coding of the data [7] and is illustrated in fig. 1b. Note, that the filters can be considered as operators projecting from one function space into another.

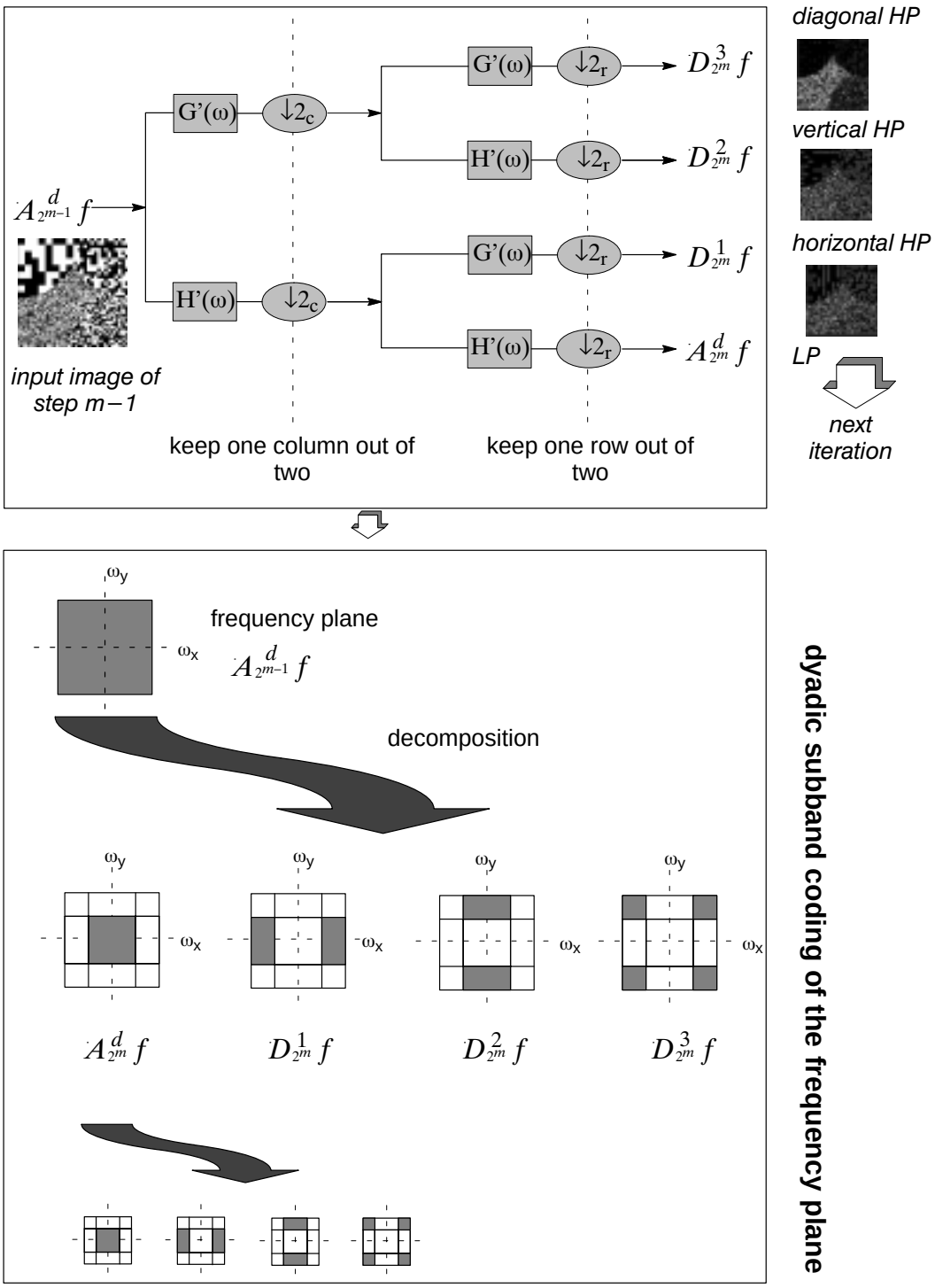

Fig. 1 a) QMF–implementation of a 2D wavelet–transform
b) Dyadic subband splitting according to [7]

As stated earlier, the WT is located both in the spatial and in the frequency plane within the boundaries of the Heisenberg principle. This property allows to adapt the level–of–detail (lod) of any reconstruction locally to interesting features and to control the approximation, neglecting unimportant or low–energy coefficients of the transform. Fig. 2 illustrates the effects of

local lod filtering for different wavelet types (see also next section). The pyramid shows especially the orientation selectivity of the different frequency channels.

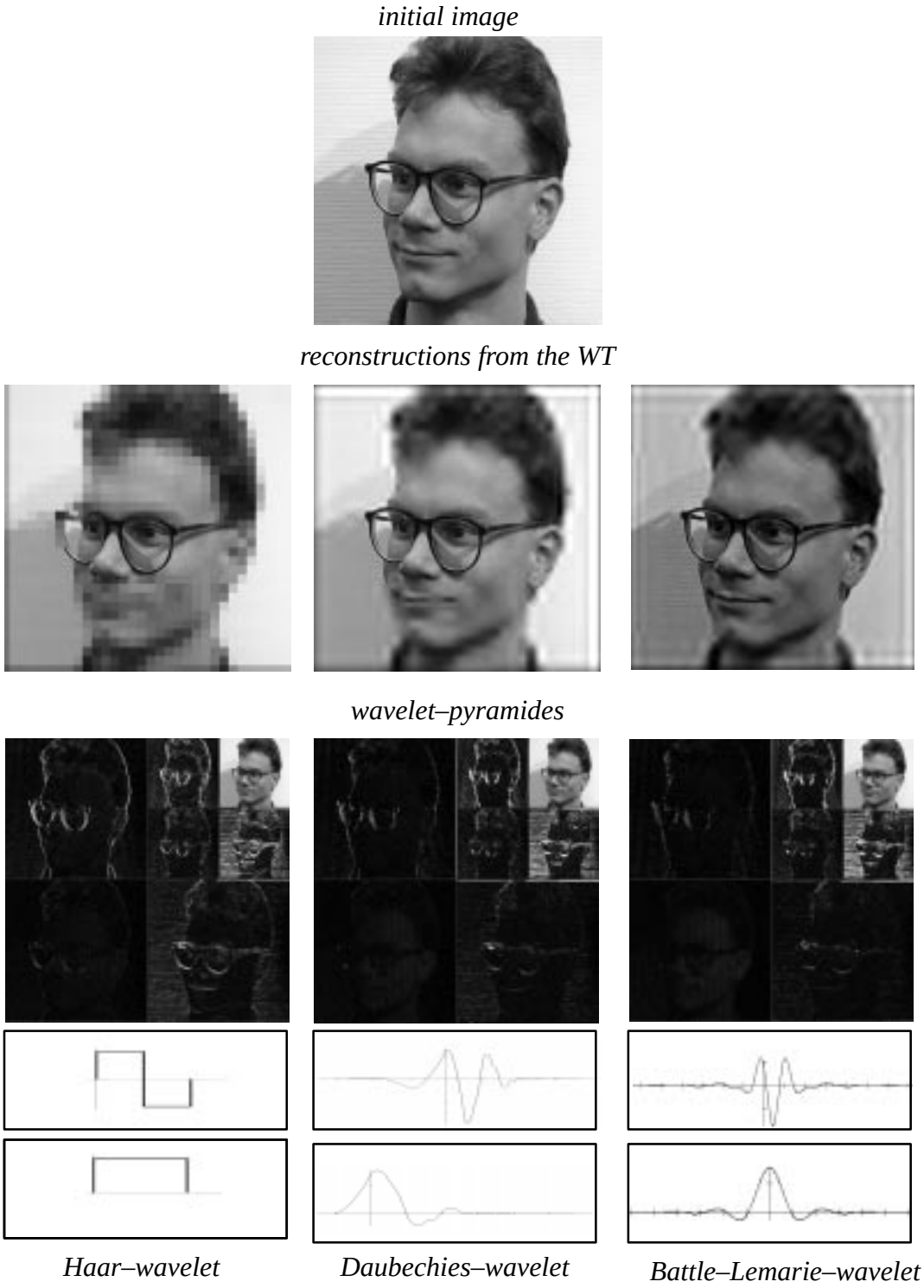

Fig. 2 Initial image, it's wavelet transforms and reconstructions with local variations of the level–of–detail for different wavelet–types

The QMF method features critical sampling of the spectrum and the WT can be formulated as a filter design problem for $H(\omega)$ where the conditions for orthogonal wavelets collapse in Fourier space to [15]:

$$
\begin{aligned}
|H(\omega)|^2 \; + \; |H(\omega \; + \; \pi)|^2 \; &= \; 1 \\
|H(0)| \; &= \; 0
\end{aligned}
\tag{12}
$$

For volumetric data sets we can set up a 3D orthogonal wavelet basis straightforwardly using the same tensor product technique. We obtain the function space V_m^3 :

$$
\begin{aligned}
V_m^3 \; &= \; V_m \otimes V_m \otimes V_m \\
V_m^3 \; &= \; \left(V_{m+1} \oplus U_{m+1}\right) \otimes \left(V_{m+1} \oplus U_{m+1}\right) \otimes \left(V_{m+1} \oplus U_{m+1}\right) \\
V_m^3 \; &= \; V_{m+1}^3 \oplus U_{m+1}^{3,1} \oplus U_{m+1}^{3,2} \oplus ... \oplus U_{m+1}^{3,7}
\end{aligned}
\tag{13}
$$

Equation (13) shows, that at each decomposition step, the space is broken up into 7 orthogonal complements that account for the principal orientations of the data, respectively. The corresponding 3D versions of the wavelets and of the scaling function can easily be derived from their one–dimensional relatives, as

$$
\begin{aligned}
\phi_{mpqr}^3(x,y,z) &:= \; 2^{-\frac{3m}{2}} \phi(2^{-m}x \; - \; p)\phi(2^{-m}y \; - \; q)\phi(2^{-m}z \; - \; r) \\
\psi_{mpqr}^{3,1}(x,y,z) &:= \; 2^{-\frac{3m}{2}} \phi(2^{-m}x \; - \; p)\phi(2^{-m}y \; - \; q)\psi(2^{-m}z \; - \; r) \\
\psi_{mpqr}^{3,2}(x,y,z) &:= \; 2^{-\frac{3m}{2}} \phi(2^{-m}x \; - \; p)\psi(2^{-m}y \; - \; q)\phi(2^{-m}z \; - \; r) \\
\psi_{mpqr}^{3,3}(x,y,z) &:= \; 2^{-\frac{3m}{2}} \phi(2^{-m}x \; - \; p)\psi(2^{-m}y \; - \; q)\psi(2^{-m}z \; - \; r) \\
\psi_{mpqr}^{3,4}(x,y,z) &:= \; 2^{-\frac{3m}{2}} \psi(2^{-m}x \; - \; p)\phi(2^{-m}y \; - \; q)\phi(2^{-m}z \; - \; r) \\
\psi_{mpqr}^{3,5}(x,y,z) &:= \; 2^{-\frac{3m}{2}} \psi(2^{-m}x \; - \; p)\phi(2^{-m}y \; - \; q)\psi(2^{-m}z \; - \; r) \\
\psi_{mpqr}^{3,6}(x,y,z) &:= \; 2^{-\frac{3m}{2}} \psi(2^{-m}x \; - \; p)\psi(2^{-m}y \; - \; q)\phi(2^{-m}z \; - \; r) \\
\psi_{mpqr}^{3,7}(x,y,z) &:= \; 2^{-\frac{3m}{2}} \psi(2^{-m}x \; - \; p)\psi(2^{-m}y \; - \; q)\psi(2^{-m}z \; - \; r)
\end{aligned}
\tag{14}
$$

The resulting data pyramid is illustrated in fig. 3. In each branch of this tree the different signal components are emphasized, being extracted from the corresponding oriented wavelet function.

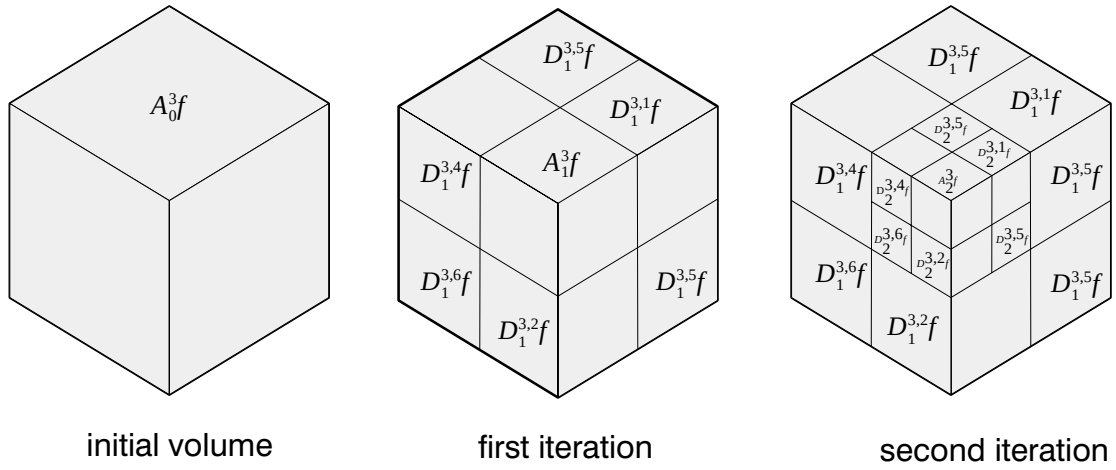

Fig. 3 Iterated decomposition of the initial volume data by the wavelet transform

Let $\tilde{c}_j \in \left\{c_{pqr}^m,\ d_{pqr}^{m,1},\ d_{pqr}^{m,2},\ d_{pqr}^{m,3},\ d_{pqr}^{m,4},\ d_{pqr}^{m,5},\ d_{pqr}^{m,6},\ d_{pqr}^{m,7}\right\}$ and $w_j(x,y,z)$ be the wavelet $\psi_{mpqr}^{3,l}$ or scaling basis function ϕ_{mpqr}^3, we write the expansion of the volume function $f(x,y,z)$:

$$f(x,y,z) = \sum_r \sum_q \sum_p c_{pqr}^M \phi_{pqr}^M + \left(\sum_m \left(d_{pqr}^{m,1}\psi_{pqr}^{m,1} + d_{pqr}^{m,2}\psi_{pqr}^{m,2} + d_{pqr}^{m,3}\psi_{pqr}^{m,3} + d_{pqr}^{m,4}\psi_{pqr}^{m,4} \right.\right.$$
$$\left.\left. d_{pqr}^{m,5}\psi_{pqr}^{m,5} + d_{pqr}^{m,6}\psi_{pqr}^{m,6} + d_{pqr}^{m,7}\psi_{pqr}^{m,7} \right)\right)$$
$$= \sum_{j=1}^{N} \tilde{c}_j \cdot w_j(x,y,z) \qquad (15)$$

N: number of voxels

Due to the immense number of basis functions (for $128^3 > 2$ Mio.) we are forced to define a significance measure for the data to reject unimportant coefficients. A generic significance for orthogonal settings can be derived by computing the local signal energy E_j from $\| \cdot \|_{L^2}$.
It is obtained for each basis as

$$E_j = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left| \tilde{c}_j\ w_j(x,y,z) \right|^2 dxdydz = |\tilde{c}_j| \cdot \| w_j(x,y,z) \|_{L^2}^2 \qquad (16)$$

Supposing the basis functions to hold

$$\left\langle w_i, w_j \right\rangle = \delta_{ij} \qquad (17)$$

we obtain the local energy by the square of the corresponding coefficients

$$E_j = \left(\tilde{c}_j\right)^2 \qquad (18)$$

The total energy E_{ges} of a 3D signal of size N is obtained by summing up all squared coefficients of the WT:

$$E_{ges} = \sum_{j=1}^{N^3} \left|\tilde{c}_j\right|^2 \tag{19}$$

Obviously, a good L^2 oracle is given by sorting and rejection of coefficients according to their magnitude.
In addition, these equations allow us to estimate the error bounds of our approximation when rejecting unimportant coefficients. Once the coefficients are filtered, we obtain a residual approximation as

$$\hat{f}(x,y,z) = \sum_{j=1}^{K} \hat{c}_j \cdot \hat{w}_j(x,y,z) \tag{20}$$

$K \leq N$

The same conditions hold for the 2D case.

2.3 Wavelet Bases for Data Representation

The elucidations above do not further restrict the mathematical properties of the wavelet bases and there have been different construction schemes proposed in literature depending on smoothness, strict local support and other criteria. This section briefly introduces the most important orthogonal wavelets. (see also [9]).

Haar Wavelets

A very simple, but discontinuous basis is given with the Haar wavelet, whose scaling function and corresponding wavelet is defined by

$$\phi(x) := \begin{cases} 1 & \text{for } 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases} \qquad \psi(x) := \begin{cases} 1 & \text{for } 0 \leq x \leq 1/2 \\ -1 & \text{for } 1/2 \leq x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

The Fourier transform of these functions (eq. (22)) show, that the expressions have optimal localization properties in the spatial domain

$$\Phi(2\omega) = e^{-i\omega} \frac{\sin(\omega)}{\omega} \qquad \Psi(2\omega) = ie^{-i\omega} \frac{\sin^2(\omega/2)}{\omega/2} \tag{22}$$

but a weak localization in frequency.

Daubechies Wavelets

In order to obtain better localization in frequency along with a minimal local support in space and smoother basis functions, Daubechies [5] proposes wavelet types as follows: the smoothness can be measured by the regularity R, (of any function ϕ) which is defined as the maximum of R such that

$$|\Phi(\omega)| \leq \frac{c}{(1+|\omega|)^{R+1}} \quad c \in \mathbb{R}^+ \tag{23}$$

This relation also describes the continuity of $\phi(x)$, where $\phi(x) \in C^i(\mathbb{R})$ with $i \leq R$. The regularity of the Daubechies wavelets is proportional to the number of vanishing moments V, defined by

$$\int_{-\infty}^{\infty} x^n \psi(x) dx = 0 \quad n \in \{0, \ldots, V-1\} \tag{24}$$

The function ϕ is only constraint by eq. (25).

$$\int_{-\infty}^{\infty} \phi(x) dx = 1 \tag{25}$$

Note, that these wavelets are given by their corresponding QMF–pairs.

Coiflet – Bases

Restricting further the scaling function to a fixed number of vanishing moments following eq. (26)

$$\int_{-\infty}^{\infty} x^n \phi(x) dx = 0 \quad n \in \{1, \ldots, V'-1\} \tag{26}$$

results in a different wavelet type, the so–called Coiflet wavelet [1]. Besides the Haar–Wavelet, strict finite support comes along in the orthonormal case with a lack of symmetry. However there is a relationship between the number of vanishing moments, stated earlier, and the symmetry of the function. Thus Coiflet bases appear to be "more symmetric" than Daubechies ones.

Battle–Lemarie Wavelets

Wavelets with an infinite support can be approximated in the frequency plane. An example with 4 vanishing moments is given by the equations (27) – (30) (see [24]).

$$\hat{\phi}(\omega) := \frac{1}{\omega^4 \sqrt{\Sigma_8(\omega)}} \tag{27}$$

where

$$\Sigma_8(\omega) := \frac{N_1(\omega) + N_2(\omega)}{105\left(\sin\frac{\omega}{2}\right)^8} \tag{28}$$

with

$$N_1(\omega) := 5 + 30\left(\cos\frac{\omega}{2}\right)^2 + 30\left(\sin\frac{\omega}{2}\right)^2\left(\cos\frac{\omega}{2}\right)^2 \tag{29}$$

and

$$N_2(\omega) := 2\left(\sin\frac{\omega}{2}\right)^4\left(\cos\frac{\omega}{2}\right)^2 + 70\left(\cos\frac{\omega}{2}\right)^4 + \frac{2}{3}\left(\sin\frac{\omega}{2}\right)^6 \tag{30}$$

The resulting wavelet type is often referred to in literature as the *Battle–Lemarie* wavelet.

Figure 4 – 6 gives again a graphical representation of the shape of some 2D wavelet basis functions. We can clearly distinguish between the smooth shape of the Battle–Lemarie wavelet on the one hand and the strict local support of the Daubechies wavelet on the other hand. It is clearly to see, where the name *"wavelet"* comes from.

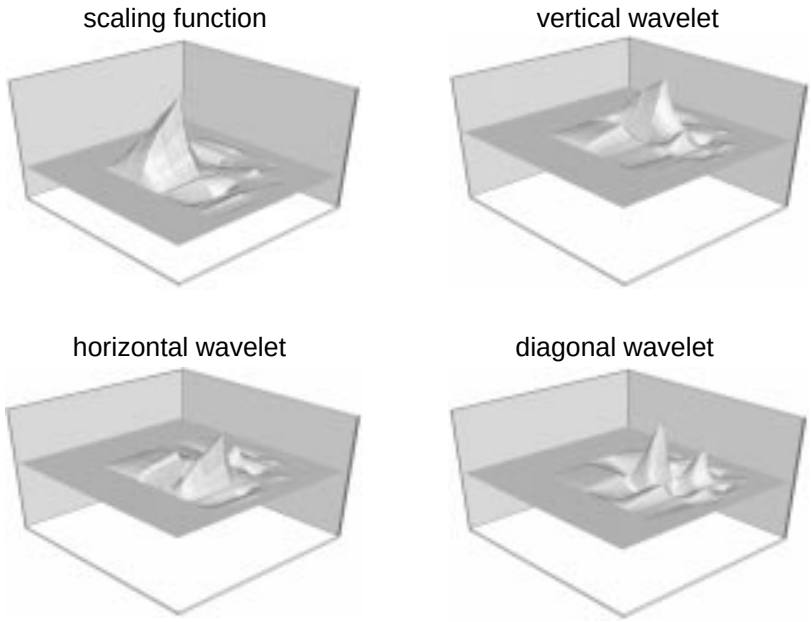

Fig. 4 2D–Daubechies 4–tap wavelets and scaling function

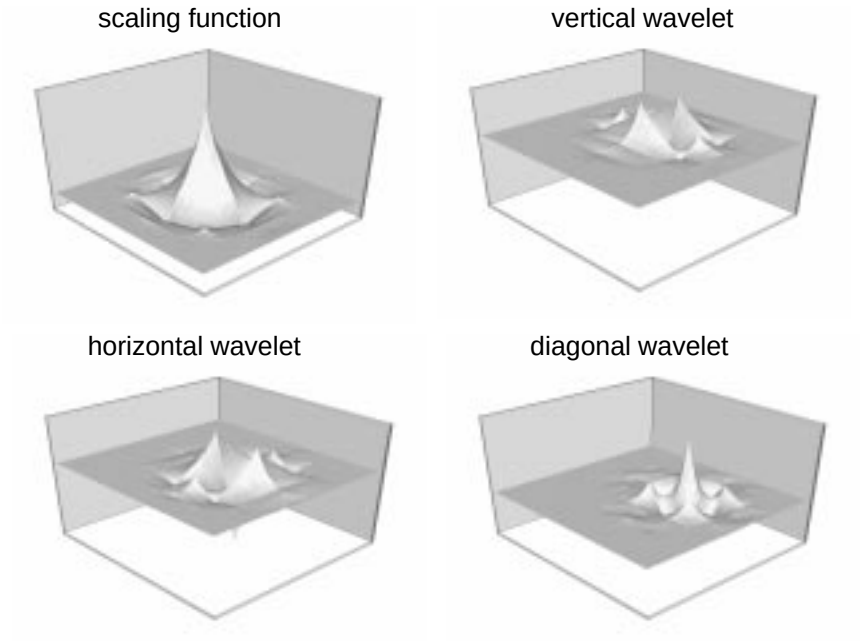

Fig. 5 2D–Coiflet wavelets and scaling function

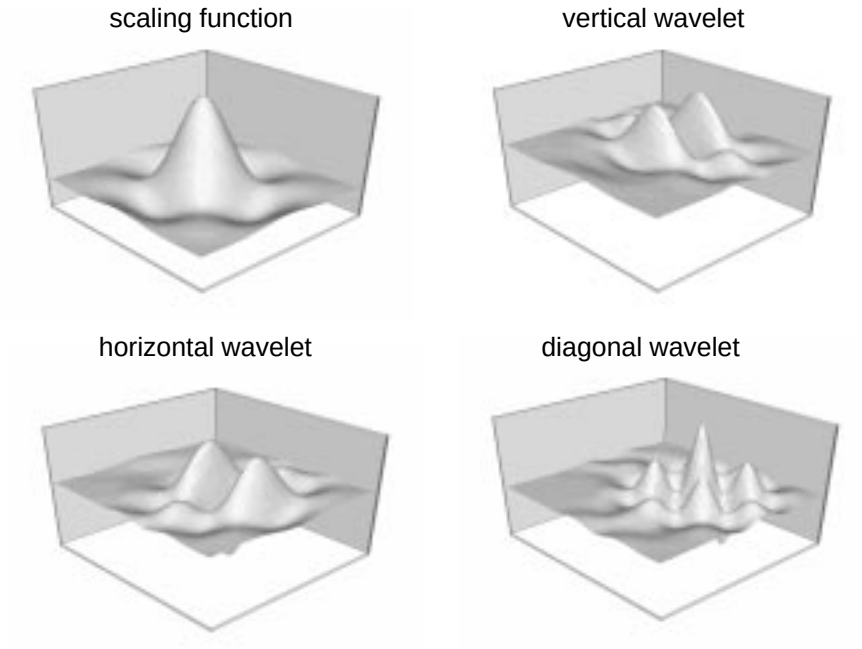

Fig. 6 2D–Battle–Lemarie wavelets and scaling function

We should also note that for continuous approximations we face the competing criteria of providing strict compact support and smooth symmetric shapes along with orthonormality in order to achieve a perfect reconstruction and compact coding. A good introduction to wavelet theory is given in [15].

3 A Framework for Integrated Data Analysis and Visualization

As stated earlier, the WT has successfully been used for data feature extraction [33],[19],[4] as well as for volume and isosurface rendering [26], [13], [10]. In particular in the field of image processing, several approaches to texture analysis have been proposed so far. Most of these techniques use a local WT on single coherent texture samples. In order to overcome this restriction, [9] proposed a new concept for texture feature extraction in images based on a global WT. The features are derived from the local wavelet coefficients and take advantage of the localization properties of the WT. These results motivated us to extend the segmentation scheme to 3D and to embed it into the rendering process – all that in the underlying data space of the WT. Figure 7 illustrates again the pipeline, where one global WT is first performed on the initial data set.

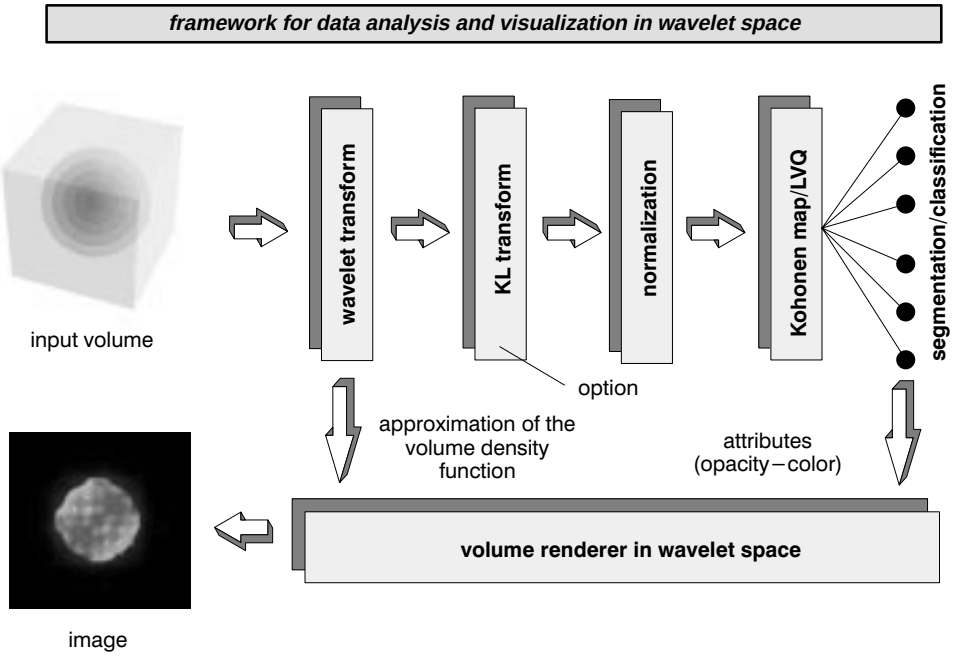

Fig. 7 Framework for integrated volume rendering and data analysis in wavelet space

Once the data is transformed, the segmentation pipeline associates a segmentation to all voxels in the data set. Therefore, we first extract a wavelet stream of coefficients from the wavelet pyramid for each voxel. This vector is taken as the feature vector describing the local properties of the data surrounding the voxel. These data have to be decorrelated, because the extraction scheme renders slightly correlated features, in spite of the orthogonality of the transform. After normalizing the features, they are feed into a neural network [8] that accomplishes clustering and classification. The result from this process is a segmentation map, that can be imported into the volume renderer. The following two chapters shed light on both data analysis and volume rendering [10] based on WTs.

4 Data Analysis in Wavelet Space

4.1 Feature Extraction in Images and Volumes

As stated earlier, the result of the wavelet transform is a pyramidal representation of the image as illustrated in figure 8b. The initial image is separated into a dyadic set of frequency channels where horizontal, vertical and diagonal components are split by the corresponding wavelets. Due to the local properties of the wavelet transform, the idea of [9] was to derive local texture features in the adjacency of any pixel (x, y) of the initial image by a set of respective coefficients from the pyramid. Supposing a depth of M in the pyramid, the set $\boldsymbol{g}(x, y) = \{g_i(x,y)\} = \{g_0,\ldots,g_{4M-1}\}$ renders a feature vector for further analysis:

$$\boldsymbol{g}_i(x,y) = \begin{cases} \left\{ d^{m,l}_{x,y} \mid l = 1 + (i \bmod 4) \right\} & \text{if } (i \bmod 4 \leq 2) \\ \left\{ c^{m}_{x,y} \right\} & \text{else} \end{cases}$$

$$m = 1 + \left\lfloor \frac{i}{4} \right\rfloor$$

Since the spatial localization of the WT decreases with lower frequencies according to the Heisenberg principle, we have to perform a bilinear interpolation between adjacent coefficients in every channel to approximate the contribution of the respective wavelet for a spatial coordinate (x, y) in the image.

Figure 8 illustrates these properties of the WT. The initial image consisting of different types of textures is transformed into the wavelet space. The result is a 4 level pyramid that separates horizontal, vertical and diagonal subbands of different spectral ranges. The local texture properties of the image can be described by the respective wavelet coefficients of the pyramid, since they provide a local spectral estimate. The contribution of the corresponding sets $\boldsymbol{g}(x, y)$ can be demonstrated by the reconstruction of the image using only coefficients of the scaling functions and of the wavelets inside the marked region. The resulting image in figure 8c features local lod, since the high frequency information is only provided locally.

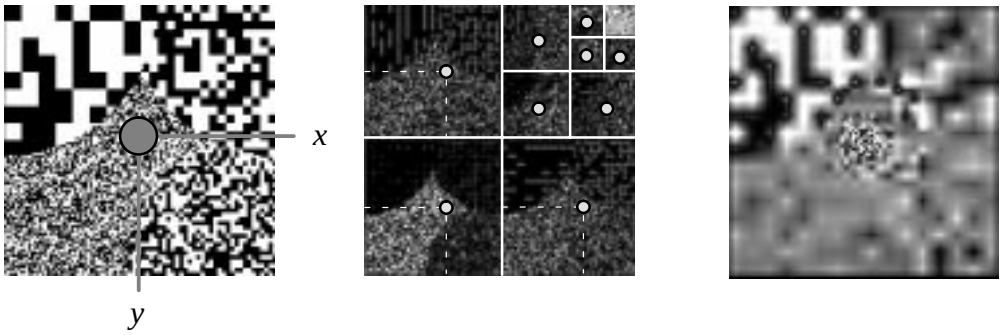

Fig. 8
a) Initial test image consisting of four different texture types and non–linear texture boundaries.
b) Subband representation by an orthogonal WT (Battle–Lemarie filter [24]).
c) Reconstruction of the image with a spatially varying resolution.

It should be noted, that the coefficients of the scaling function c_{xy}^{m} of each resolution step m are added into our feature vector. They represent the low–pass parts of the signal and are decomposed further. This gives rise to a correlation of the individual features, but it turned out to provide better results. For this end, decorrelation is a meaningful step in the pipeline even when employing orthonormal wavelet types.

This scheme can be extended into 3D as shown in fig. 9. The local volume texture properties surrounding a voxel (x,y,z) are described by their respective set of coefficients $\boldsymbol{g}(x,y,z)$ from the volume pyramid.

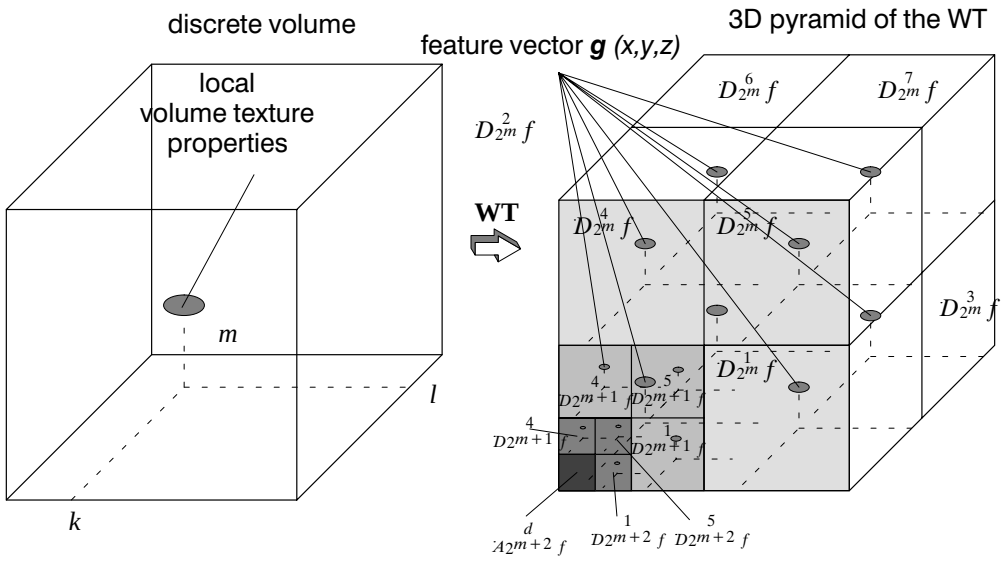

Fig. 9 Illustration of the extraction of local volume texture features from the volume pyramid.

4.2 A Pipeline for Data Analysis

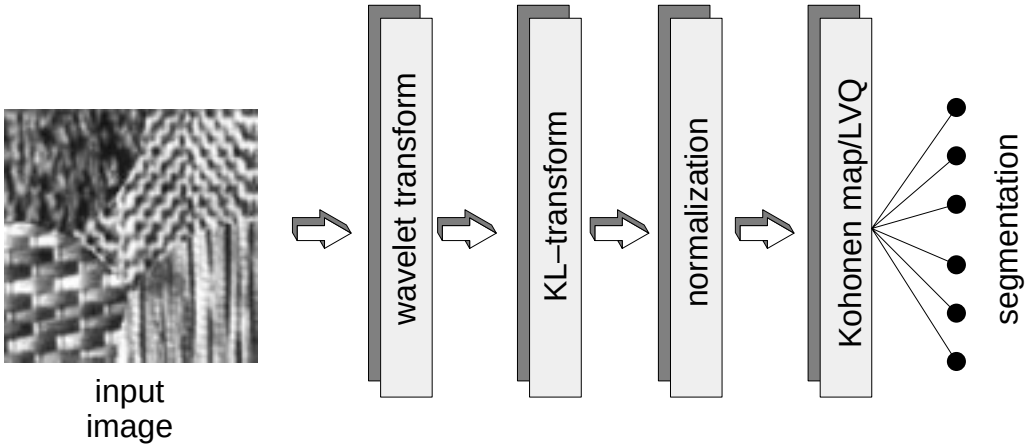

Fig. 10 Wavelet–based data analysis pipeline.

Figure 10 shows again the pipeline for data analysis as first applied in [9] and later extended in [11]. After an initial WT decorrelation of the coefficients is required to provide an optimal representation of the extracted features. There are many different approaches in mathematical statistics to perform this. The method proposed here employs the PCA [7]. Since the amplitude distribution of the data is unknown, the normalization step supposes an uniform distribution and scales the data straightforwardly to its minimum and its maximum.

The required cluster analysis of the texture features as well as the supervised classification step is accomplished with an extension of the Kohonen Feature Map proposed by [8]. The basic topology of the network is illustrated in figure 11. The competitive layer is extended to 3D and each neuron j is assigned to a specific color triplet (R_j, G_j, B_j) in the RGB color space. During the non–supervised organization process the network performs a C–means clustering of the incoming

data, where each weight vector $\boldsymbol{w}_j$ represents the position of a centroid in feature space. Since the network tries to preserve the data topology, neighbored neurons in the output layer are mapped to neighbored cluster centers in feature space. Visualizing these cluster centers in terms of color results in a mapping of a multidimensional data distribution to color similarities in RGB. The updating of the weights $\boldsymbol{w}_j$ for each neuron j during learning follows the basic rules of Kohonen:

$$\Delta w_{ij} \;=\; \alpha(t) \cdot (g_i\,(x,y) - w_{ij}) \;, \quad \text{if neuron } j \in N_j(t) \tag{31}$$

$\alpha(t)$: time dependent learning rate

g_i: feature coordinate i

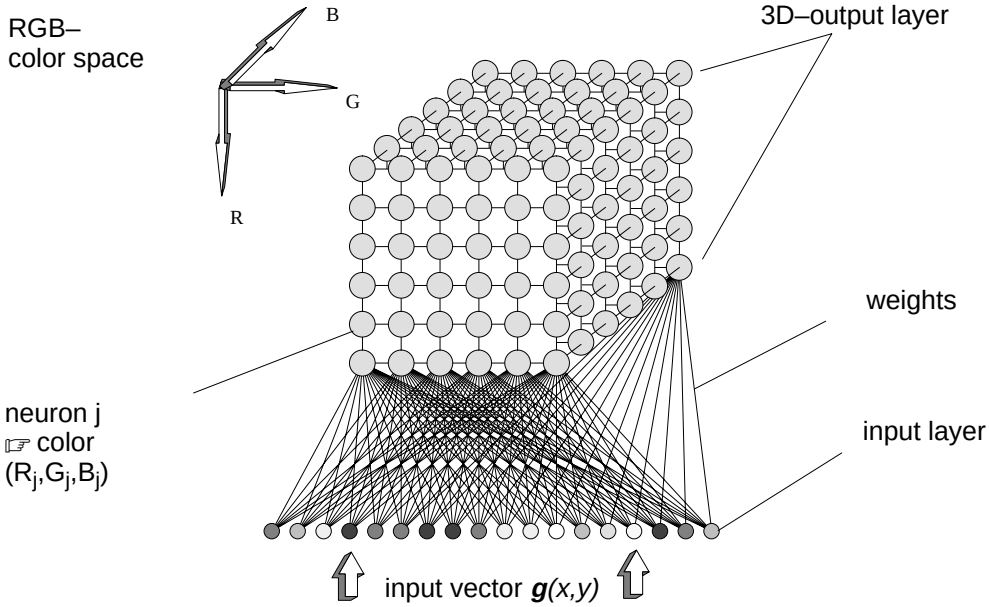

Fig. 11 3D Kohonen map and color assignment

$N_j(t)$ represents the local neighborhood of neuron j with the minimum Euclidian distance to the feature vector $\boldsymbol{g}(x,y)$.

4.3 Some Results on Texture Analysis

Figure 12 (upper) shows four different images generated to investigate the capabilities of the approach. The task was to perform a classification of the textures in the different images. The following wavelets were employed:

– Gabor wavelets (8 spatial orientations)
– Daubechies wavelets (4–tap)
– Haar wavelets
– Battle–Lemarie wavelets

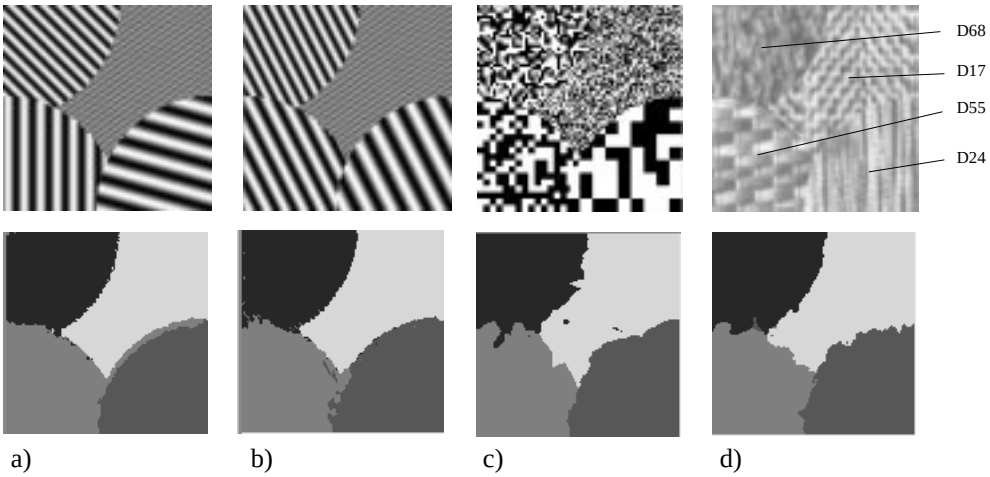

Fig. 12 Different texture compositions and results:
a) Sine1: Sine gratings of different spectral ranges and orientations
b) Sine2: Sine gratings of different spectral ranges
c) Random dots
d) Brodatz textures: D17, D24, D55, D68

Figure 13 illustrates the error we obtained for different images and wavelets as a function of the iteration depth. The results show that the accuracy depends on the texture type. It is interesting to compare the homogeneity of the results within the texture regions. The errors recorded appear mostly at the boundaries.

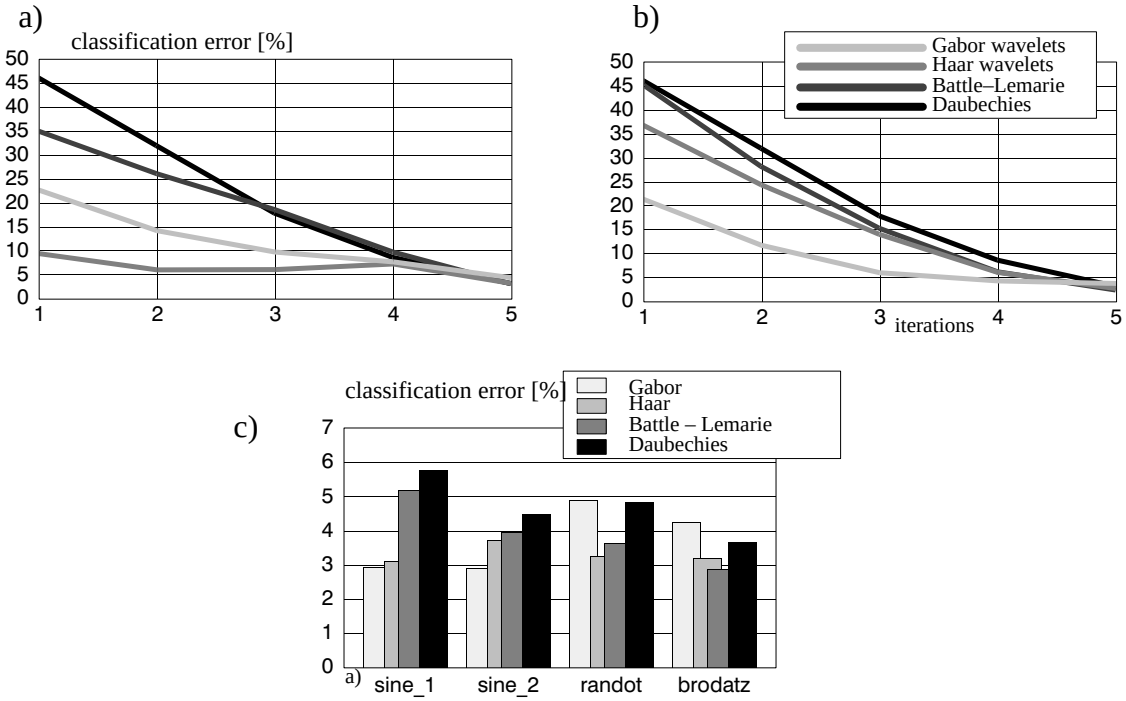

Fig. 13 Results for a) RANDOT
b) BRODATZ
c) Best matches for all samples

A detailed analysis of the performance of this algorithm for image texture analysis is provided in [9]. The following section of the paper elaborates a volume visualization method based on wavelets according to [10].

5 Volume Rendering in Wavelet Space

5.1 Preliminary Remarks

The basic goal of volume rendering is to find a good approximation of the low albedo volume rendering integral [2], [16] which expresses the relation between the volume intensity and opacity functions and the intensity in the image plane. Most of the standard volume rendering algorithms therefore approximate an integral equation with the ray parameter t of the type

$$I = \int_{t_1}^{t_2} C(t)\ e^{-\int_{t_1}^{t} \alpha(s)ds}\ dt \qquad (32)$$

where *C(t)* stands for a volume intensity function and includes emitted, scattered and reflected light. $\alpha(t)$ denotes the opacity function of the data and can be used to encode data features to be enhanced in the final images. Hence, the inner integral includes the self–occlusion of the volume.

The most common way to get a numeric solution of eq. (32) employs a first–order quadrature of the inner integral along with a linearization of the exponential. The outer integral is also solved by a finite sum of uniform samples. We yield

$$I = \sum_{k=1}^{M} C_k\ \alpha_k \prod_{i=1}^{k-1} (1 - \alpha_i) \qquad (33)$$

where α_k are the opacity samples along the ray and C_k are the local color values derived from the illumination model. Note that I has to be computed for each spectral sample λ, i.e. in R,G,B.

A good mathematical analysis of the problem and error bounds of numeric quadrature is provided in [28].

5.2 Approximate Solutions of the Volume Rendering Equation

In chapter 2 we elaborated that some wavelet types are not given in a closed form solution and properties of smoothness, symmetry and the keeping of the orthonormality can only be achieved by infinite support. That is, explicit approximations are vulnerable to truncation errors. In order to compare different wavelet types and to be independent of the basis function, we require a generic continuous representation scheme. Furthermore, the final goal is to find an approximate and analytic representation for the ray intensity function. Thus the discrete values of the wavelet functions obtained by feeding Dirac–pulses into the filter bank are interpolated by piecewise polynomials using cubic splines in each interval $I \subseteq \mathbb{R}^3$.

Note, that this approach provides a good framework for fundamental studies and investigations, however, it is not suited for building fast rendering schemes.

From the sections above we know, that a 3D wavelet decomposition of a volume data set accomplishes an expansion of the volume function *f(x,y,z)* according to eq. 15.

The separability of the bases $\hat{w}_j$ allows us to write eq. 20 as

$$\hat{f}(x,y,z) = \sum_{j=1}^{K} \hat{c}_j \; b_j^1(x) \; b_j^2(y) \; b_j^3(z) \qquad (34)$$

where the b_j represent the *x,y* and *z*–components of the tensor product basis functions.

The image generation with ray casting turns out to be a parametrization of the ray as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \alpha_x t + \beta_x \\ \alpha_y t + \beta_y \\ \alpha_z t + \beta_z \end{pmatrix} \qquad \alpha = \left[\alpha_x, \alpha_y, \alpha_z\right]^T, \quad \beta = \left[\beta_x, \beta_y, \beta_z\right]^{T,} \qquad (35)$$

α and β are the viewing direction and the eyepoint respectively.

We obtain the intensity function along the ray with

$$\hat{f}(t) = \hat{f} \begin{pmatrix} \alpha_x t + \beta_x \\ \alpha_y t + \beta_y \\ \alpha_z t + \beta_z \end{pmatrix} = \sum_{j=1}^{K} \hat{c}_j \; b_j^1(\alpha_x t + \beta_x) \; b_j^2(\alpha_y t + \beta_y) \; b_j^3(\alpha_z t + \beta_z) \qquad (36)$$

This scheme is illustrated in fig. 14. The ray intensity function is provided by projecting the individual basis functions onto the ray and by superimposing them. It is accomplished by scaling with α and by translating with β. Due to the piecewise spline interpolation we get now a continuous approximation of the ray intensity function. In each interval $\left[t_n^j, t_{n+1}^j\right]$ and for each component of the resulting vector the cubic polynomials $b_j^{i,n}(t)$ are given for each wavelet w_j as monomials

$$b_j^{i,n}(t) = \sum_{k=0}^{3} \tilde{a}_{k,j}^{i,n} \; t^k \qquad i = 1,2,3 \qquad (37)$$

and their coefficients $\tilde{a}_{k,j}^{i,n}$ as

$$\tilde{a}_{k,j}^{i,n} = \sum_{l=k}^{3} \binom{l}{l-k} a_{l,j}^{i,n} \; \alpha^{\,k} \; \beta^{\,l-k} \qquad (38)$$

$a_{l,j}^{i,n}$: Spline coefficients in interval $[t_n^j, t_{n+1}^j]$

Thus we can write

$$b_j^{i,n}(t) = \sum_{k=0}^{3} \left[\sum_{l=k}^{3} \binom{l}{l-k} \; a_{l,j}^{i,n} \; \alpha^{\,k} \; \beta^{\,l-k} \right] t^{\,k} \quad with \;\; t \in [t_n^j, t_{n+1}^j] \qquad (39)$$

The final expression for $\hat{f}(t)$ is obtained by

$$\begin{aligned} \hat{f}(t) = \sum_{j=1}^{K} \hat{c}_j & \left[\sum_{k=0}^{3} \left[\sum_{l=k}^{3} \binom{l}{l-k} a_{l,j}^{1,n} \; \alpha_x^{\,k} \; \beta_x^{\,l-k} \right] t^{\,k} \right] \\ & \cdot \left[\sum_{k=0}^{3} \left[\sum_{l=k}^{3} \binom{l}{l-k} a_{l,j}^{2,n} \; \alpha_y^{\,k} \; \beta_y^{\,l-k} \right] t^{\,k} \right] \qquad t \in [t_n^j, t_{n+1}^j] \\ & \cdot \left[\sum_{k=0}^{3} \left[\sum_{l=k}^{3} \binom{l}{l-k} a_{l,j}^{3,n} \; \alpha_z^{\,k} \; \beta_z^{\,l-k} \right] t^{\,k} \right] \end{aligned} \qquad (40)$$

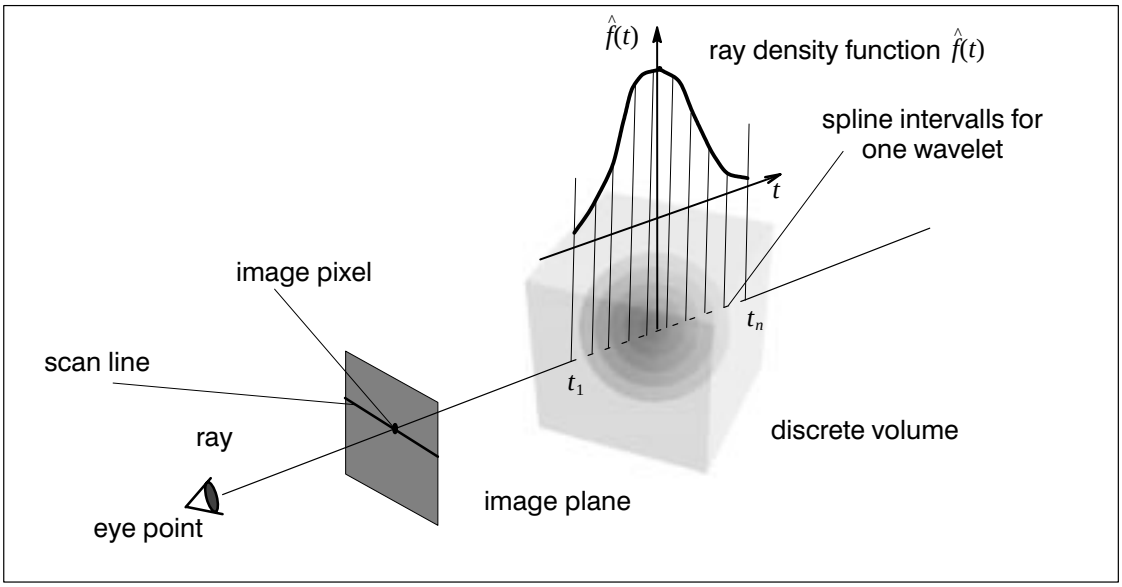

Fig. 14 Illustration of the rendering process

Note, that eq. (40) finally provides an approximation with cubic volume splines which can be integrated straightforwardly. Unfortunately, however, the rendering equation incorporates an exponential absorption term.

Now, depending on the application there are different ways to compute the image intensity *I*. In the simple, but important [32] case of constant transfer functions eq. (32) simplifies to

$$I = \int_{t_1}^{t_2} f(t)\ dt \tag{41}$$

We obtain a piecewise analytic solution for each interval $\left[t_n^j, t_{n+1}^j\right]$ by the piecewise primitive functions $W_j^n(t) = \int b_j^1(t) \cdot b_j^2(t) \cdot b_j^3(t)\ dt$, with

$$I = \sum_{j=1}^{K} I_j = \sum_{j=1}^{K} \hat{c}_j \sum_{n=0}^{L^j-1} \left[W_j^n(t_{n+1}^j) - W_j^n(t_n^j) \right] \quad L^j : \text{number of spline intervalls} \tag{42}$$

Note, that the size and number of the intervals L^j depend either on how the viewing ray intersects the wavelet and on the iteration *m*. In the case of a close form representation of the $W_j(t)$, such as with B–spline wavelets the relations collapse to:

$$I = \sum_{j=1}^{K} I_j = \sum_{j=1}^{K} \hat{c}_j \left[W_j(t_2) - W_j(t_1) \right] = \sum_{j=1}^{K} \hat{c}_j \int_{-\infty}^{\infty} \tilde{w}_j(t) dt \tag{43}$$

The upper integral can be computed effectively using Fourier projection slicing [22].

In order to include a self–occlusion term and to evaluate the inner integral, we expand the function $\alpha(s)$ by our bases and obtain the following expression:

$$e^{-\int_{t_1}^{t} \alpha(s) ds} = e^{-\sum_{j=1}^{K} \alpha_j \sum_{n=0}^{L^j(t)-1} \left[W_j^n(t_{n+1}^j) - W_j^n(t_n^j) \right]} \qquad \forall j\ ,\ t_0^j = t_1 \quad t_{L^j(t)}^j = t \tag{44}$$

α_j: wavelet coefficient for $\alpha(x,y,z)$

Due to the local support of the WT, we have only to account a subset of wavelets along the ray. A linear approximation of the exponential function aligned to the spline intervals yields

$$e^{-\int_{t_1}^{t} \alpha(s) ds} \approx \prod_{j=1}^{K} \alpha_j \prod_{n=0}^{L^j(t)-1} \left(1 - \left[W_j^n(t_{n+1}^j) - W_j^n(t_n^j) \right]\right) \tag{45}$$

Again, it has to be noted that a closed form representation provides a compact approximation, but worse error bounds.

$$e^{-\int_{t_1}^{t} \alpha(s)ds} \approx \prod_{j=1}^{K} \alpha_j \left(1 - \left[W_j(t) - W_j(t_1)\right]\right) \qquad (46)$$

The final discrete solution of the rendering equation depends on the shading model. The gradient $\nabla f(x,y,z)$ that is required for shading can be computed easily from

$$\nabla \hat{f}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \sum_{j=1}^{K} \hat{c}_j \left(\frac{d}{dx} b_j^1(x)\ b_j^2(y)\ b_j^3(z)\ ,\ b_j^1(x)\ \frac{d}{dy} b_j^2(y)\ b_j^3(z)\ ,\ b_j^1(x)\ b_j^2(y)\ \frac{d}{dz} b_j^3(z) \right) \qquad (47)$$

We have to mention here, that the implementation of these equations is not straightforward. Although the piecewise spline interpolation of the basis functions has only to be performed for the 1D–prototypes of ψ and ϕ, the calculation of the spline intervals in t are computationally very expensive.

The problem of getting isosurfaces in the images will be treated separately and discussed in the next section. The pictures in this paper are based on eq. (41).

5.3 Examples

In fig. 15 a Gaussian density distribution was voxelized at a resolution of 32^3 and rendered with different numbers of coefficients and iteration depths. The isosurface was set to $\tau = 0.5$, the intensities below that threshold are represented as a bluish translucency. We should note that symmetry and shape of the wavelet strongly influence the shape of the isosurface. Asymmetric and fractallike functions, as for instance the Daubechies wavelet, generate artifacts, such as rips or modulations of the translucency. It is interesting to compare them to the shapes obtained by a standard marching cubes on the initial volume data set.

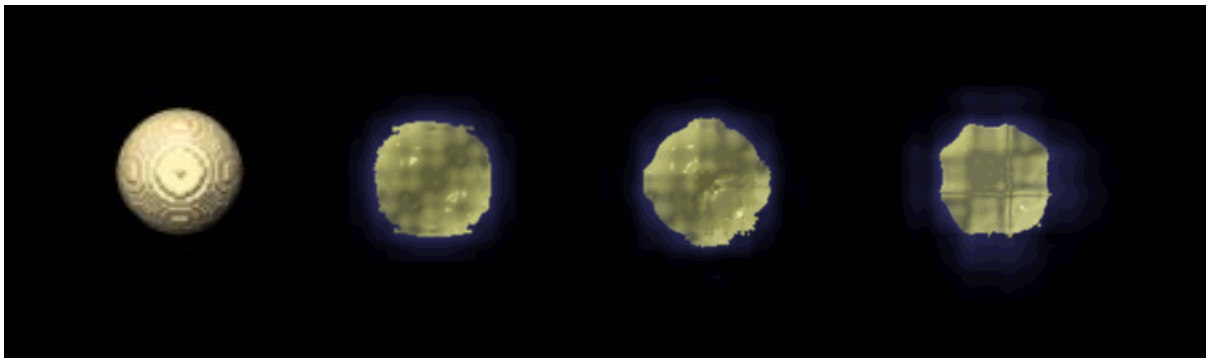

Fig. 15 Images obtained from a Gaussian density distribution of 32^3 voxels:
(a) Isosurface at $\tau = 0.5$ with a marching cubes algorithm.
(b) Isosurfaces and translucent hull obtained from a Battle–Lemarie wavelet with 961 coefficients.
(c) Isosurfaces and translucent hull obtained from a Coiflet wavelet with1006 coefficients.
(d) Isosurfaces and translucent hull obtained from a Daubechies wavelet with1154 coefficients.

The pictures were generated with our hybrid renderer that composes the volumetric intensities with the isosurfaces. In these examples, the isosurfaces were rendered using the implicit function approach of [17]. The coefficients were filtered according to the significance measures in section 2.2.

6 Isosurface Reconstruction

6.1 Implicit Function Approach versus Marching Cubes

The wavelet decomposition offers different ways to get isosurfaces from volume data. Direct volume integration to get opaque surfaces, as proposed by [21] or [6] usually base on shading models that estimate the surface normal by means of the volume gradient ∇f. Another way to solve the isosurface problem is proposed by [26]. It bases on the idea, that isosurfaces of a threshold τ satisfy

$$\hat{f}(x,y,z) = \tau \;\Rightarrow\; \hat{f}(x,y,z) - \tau = 0 \qquad (48)$$

in the continuous approximation of eq. (20).

Equation (48) leads to an implicit function that can be rendered using methods as in [17]. Detailed descriptions of how to compute the Lipschitz condition are given in [10].

Unfortunately, there are several shortcomings in this approach: Due to the huge number of basis functions, Kalra's method becomes extremely time consuming. Furthermore, the appearance of the isosurface is strongly influenced by the shape of the wavelet, which has to be represented in a continuous form. Hence, it becomes interesting to apply a simple marching cubes technique [23] on the expanded data set and to compose the polygons with the volume data during the rendering process. In these cases, the isosurface generation is no longer performed in wavelet space, but the local data quality is still controlled by the wavelets.

6.2 Examples

Figure 16a shows the bust of Johann Strauss, as it is derived from a 3D laserscanner. This model was illuminated and voxelized with a resolution of 32^2x64 voxels. The isosurfaces obtained from a marching cubes are presented for a flat shaded reconstruction in fig. 16b respectively. Figures 17a,c,b show the isosurface reconstruction with Kalra's method [17] which is accomplished on immediately by ray tracing the basis functions in wavelet space. In figure 18a,b,c the same reconstructions are presented after an inverse transform of the filtered data and with a marching cubes algorithm. In both cases, an increasing number of coefficients were employed to encode the data. We can clearly recognize that the level of detail increases as the number of wavelets is raised.

Fig. 16 a) Range data of a bust of Johann Strauss
b) Voxelized volume data at a resolution $32^2 \times 64$
c) Isosurface reconstruction using a marching cubes algorithm
(range data courtesy provided by the ZGDV, Darmstadt, Germany)

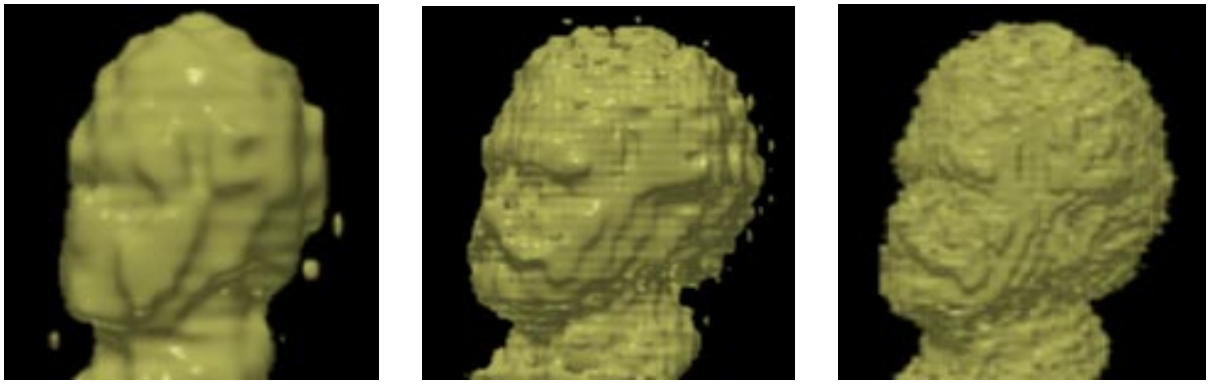

Fig. 17 Isosurface reconstruction using Kalra's method and 3D–wavelets with τ=0.5
a) 1180 coefficients
b) 2832 coefficients
c) 9601 coefficients

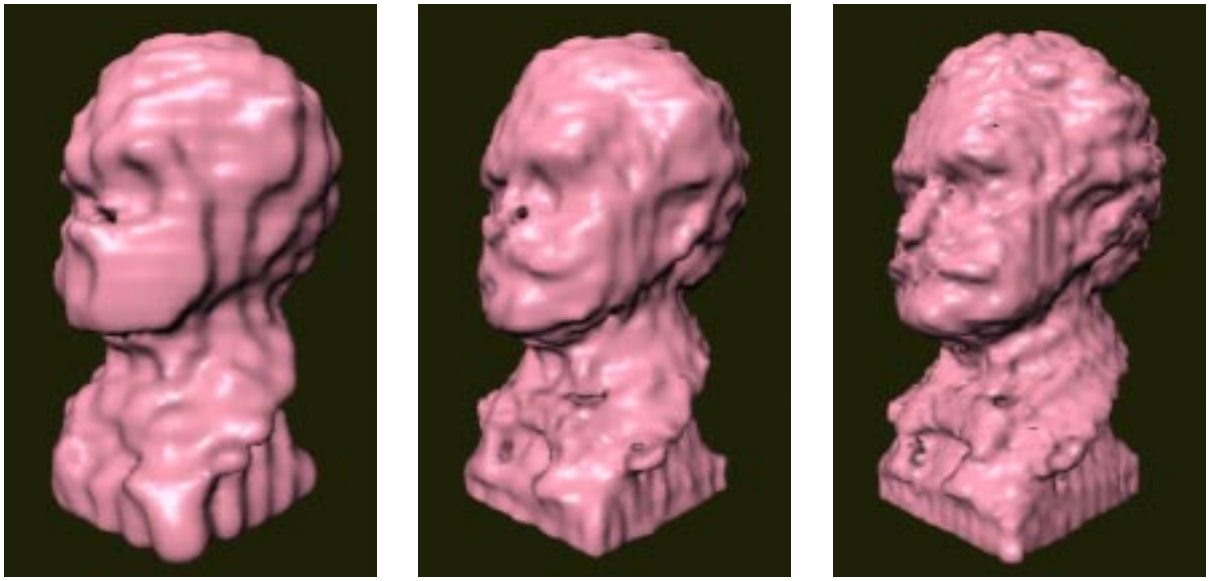

Fig. 18 Isosurface reconstruction using marching cubes methods with τ=0.5 and Battle–Lemarie–wavelets
a) 1180 coefficients
b) 2832 coefficients
c) 9601 coefficients

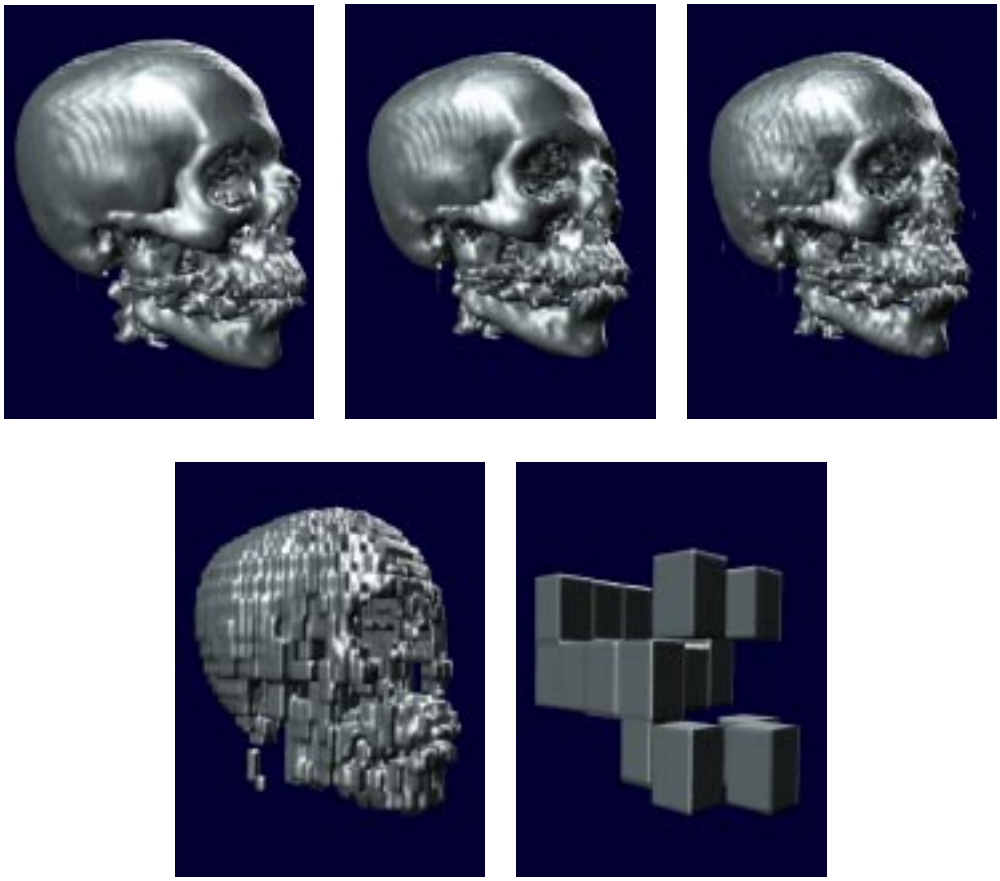

Fig. 19 Isosurfaces from a human scull obtained by Haar decompositions of the data with different levels of approximation:
a) 100 % coefficients

b) 15.5 % coefficients
a) 6.8 % coefficients
a) 0.3 % coefficients
a) 0.02 % coefficients

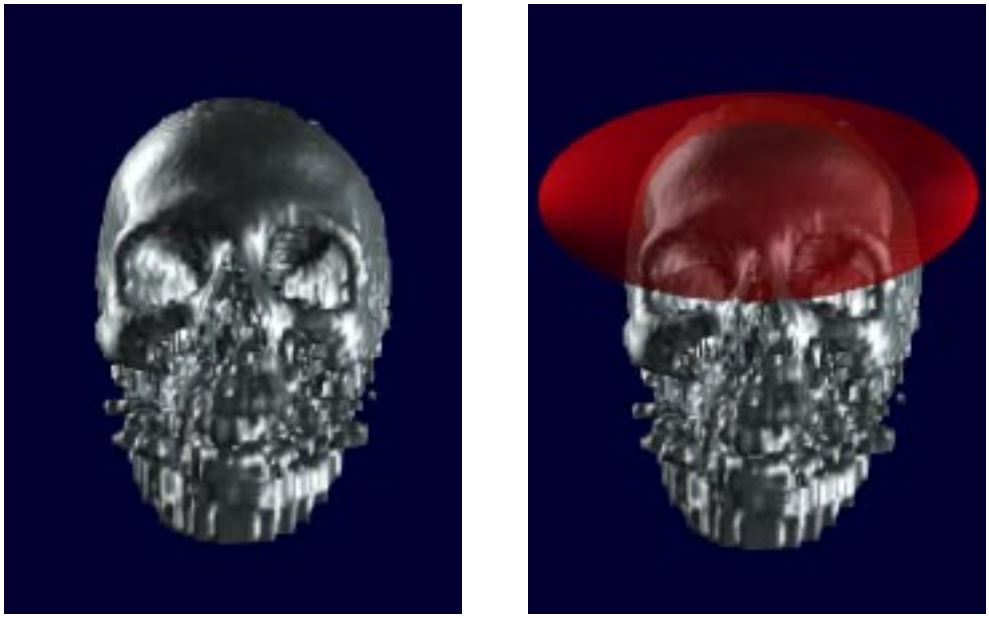

Fig. 20 Contolling the local level–of–detail of the data by Gaussian weighting functions:
a) Result obtained with Haar wavelets.
b) Illustration of the weighting function (red ellipsoid).

In fig. 19 a series of isosurface reconstructions from a human scull is depicted with an increasing number of coefficients. Due to the Haar wavelets employed for the decomposition the surface becomes more or less ”box–like”. Figure 20 illustrates the localization properties of the WT. Here, a 3D Gaussian weighting–function (transparent ellipsoid in fig. 20b) was applied in wavelet space to enhance the approximation. Obviously, we end up with a perfect reconstruction of the isosurface in those spatial regions affected by the Gaussians.

7 Conclusion and Future Research

In this paper, a new framework for integrated volume rendering and data analysis in wavelet space was elaborated. It was shown, that the WT is well suited for the extraction of local data features in images. Future research activities have to focus on testing the method on volume data sets and to combine it with current imaging techniques, such as morphological processing. Furthermore the proposed rendering method provides piecewise analytic solutions of the intensity integral, since the underlaying volume is approximated continuously by polynomials. The quality of the results depends strongly on the types of the selected basis function. Yet, it turns out, that the projection of the bases onto the ray is computationally expensive. Hence we have to find a wavelet, that is smooth, of strict finite support, orthonormal and that provides a closed–form integral in t. These properties account for both rendering and data analysis, but cannot be satisfied in common. Thus a compromise has to be found as in terms of biorthogonal wavelets.

8 Acknowledgement

Most of this research was done at the Computer Graphics Center in Darmstadt, Germany. It was supported in parts by the German Telekom within the KAMEDIN project. The author thanks

Lars Lippert, Rolf Koch, Oliver Staadt and Andy Dreger for implementing parts of the presented methods during their Diploma theses.

9 Literature

[1] Akansu, A. N., Haddad, R.A.: ”Multiresolution Signal Decomposition”. Academic Press, Inc., 1992

[2] Blinn, J., F.: ”Light Reflection Functions for Simulation of Clouds and Dusty Surface”. Computer Graphics, Vol. 16, No. 3, pp. 116 – 123, 1993

[3] Bovik, A. C.; Clark, M.; Geisler, W.: ”Multichannel Texture Analysis Using Localized Spatial Filters”. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 1, pp. 55 – 73, 1990

[4] Chang, T.; Kuo, C.: ”Texture Analysis and Classification with Tree–Structured Wavelet Transform”. IEEE Transactions on Image Processing, Vol. 2, No. 4, pp. 429 – 441, 1993

[5] Daubechies, I.: ”The Wavelet Transform, Time–Frequency Localization and Signal Analysis”. IEEE Transactions on Information Theory, Vol. 36, pp. 961 – 1005, 1990

[6] Drebin, R.A.; Carpenter, L.; Hanrahan, P. : ”Volume Rendering”. Computer Graphics, Vol. 22, No. 4, pp. 125–134, 1988

[7] Groß, M. : ”Visual Computing”. Springer–Verlag, 1994

[8] Groß, M.; Seibert, F.: ”Visualization of Multidimensional Image Data Sets using a Neural Network”. The Visual Computer, Vol. 10, pp. 145 – 159, 1993

[9] Groß, M.; Koch, R; Lippert, L.; Dreger, A.: ”Multiscale Image Texture Analysis in Wavelet Space”. IEEE–ICIP’94, to appear 1994

[10] Groß, M.; Lippert, L.; Dreger, A.; Koch, R.: ”A New Method to Approximate the Volume Rendering Equation Using Wavelets and Picewise Polynomials”, Comptuers& Graphics, Vol. 19, No. 1, to appear, 1995

[11] Groß, M.; Koch, R.: ”Visualization of Multidimensional Shape and Texture Features in Laser Range Data Using Complex–Valued Gabor Wavelets”. IEEE Transactions on Visualization and Computer Graphics, Vol. 1 No. 1, pp. 44 – 59, 1995

[12] Grossmann, A.; Morlet, J.: ”Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape”. SIAM Journal of Mathematical Analysis, Vol. 15, pp. 723–736, 1984

[13] He, Tao; Wang, S.; Kaufman, A.: ”Wavelet–Based Volume Morphing”. ACM Volume Rendering Symposium, to appear 1994

[14] Höhne, K.H.; Bomas, M.; Pommert, A.; Riemer, M.; Schiers, C.; Tiede, U.; Wiebecke, G.: ”3D Visualization of Tomographic Volume Data using Generalized Voxel Model”. The Visual Computer, Vol.6, No. 1, pp. 28–36, 1990

[15] Jawerth, B.; Sweldens, W.: ”An Overview of Wavelet Based Multiresolution Analyses”, Department of Mathematics, University of South Carolina, internal report

[16] Kajiya, J.T.; Von Herzen, B.P.: ”Ray Tracing Volume Densities”. SIGGRAPH ’84, pp. 165–174, 1984

[17] Kalra, D.; Barr, A.H.: ”Guaranteed Ray Intersection with Implicit Surfaces”. Computer Graphics (Proc. SIGGRAPH ’89), Vol. 23, pp. 297–306, 1989

[18] Krueger, W.: ”The Application of Transport Theory to Visualization of 3D Scalar Data Fields”. IEEE, pp. 273–279, 1990

[19] Laine, A.; Fan, J.: ”Texture Classification by Wavelet Packet Signatures”. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 11, pp. 1186 – 1191, 1993

[20] Laur, D.; Hanrahan, P.: ”Hierarchical Splatting: A Progressive Refinement Algorithm for Volume Rendering ”. Computer Graphics and Applications, Volume 25, Number 4, pp. 285–288, 1991

[21] Levoy, M.: ”Display of Surfaces from Volume Data”. IEEE Computer Graphics and Applications, Volume 8, Number 5, pp. 29–37, 1988

[22] Lippert, L.; Groß, M.: ”Fast Wavelet–Based Volume Rendering by Accumulation of Transparent Texture Maps”. to apprear Eurographics 95 Proceedings, Computer Science Department Internal Report No. 228

[23] Lorensen, W.E.; Cline, H.E.: ”Marching Cubes: A High Resolution 3D Surface Construction Algorithm”. Computer Graphics, Vol. 21, pp. 163–196, 1987

[24] Mallat, S.: ”A Theory for Multiresolution Signal Decomposition: The Wavelet Representation”. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, pp. 674–693, July 1989

[25] Meinzer, H.P.; Meetz, K.; Scheppelmann, D.; Engelmann, U.; Baur, H.J.: ”The Heidelberg Ray Tracing Model”. IEEE Computer Graphics and Application, Nov. , pp. 34–43, 1992

[26] Muraki, S.: ”Volumetric Shape Description of Range Data using ’Blobby Model’”. Computer Graphics, Vol. 25, No. 4, pp. 227–235, 1991

[27] Muraki, S.: ”Multiscale 3D Edge Representation of Volume Data by a DOG Wavelet”. Symposium on Volume Visualization, to appear 1994

[28] Novins, K.; Arvo, J.: ”Controlled Precision Volume Integration”. ACM Workshop on Volume Visualization, pp. 83–89, 1992

[29] Sabella, P.: ”A Rendering Algorithm for Visualizing 3D Scalar Fields”. Computer Graphics, Vol. 22, No.4, pp. 160–164, 1988

[30] Sarkas, G.; Gerth, M.: ”Sampling and Anti–Aliasing of Discrete 3D Volume Density Textures”. Eurographics, Vienna, Sep., pp. 87–102, 1991

[31] Simoncelli, E. P.; Adelson, E.H.: ”Non–Separable Extensions of Quadrature Mirror Filters to Multiple Dimensions”. Proceedings of the IEEE, Vol. 78, No. 4, pp. 652–664, 1990

[32] Totsuka, T., Levoy, M.: ”Frequency Domain Volume Rendering”. Computer Graphics Proceedings, Annual Conference Series, pp. 271–278, 1993

[33] Unser, M.; Eden, M.: ”Multiresolution Feature Extraction and Selection for Texture Segmentation”. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, pp. 717 – 728, 1989

[34] Vetterli, M.: ”Wavelets and Filter Banks: Theory and Design”. IEEE Transactions on Signal Proc., Vol. 40, pp. 2207 – 2231, 1992